

Assignment 1

In this assignment ~~we~~ you will create a bookstore-application. In the bookstore not only books are sold, but also magazines and CDs.

Create an abstract class “BookStoreItem” and give it the following properties:

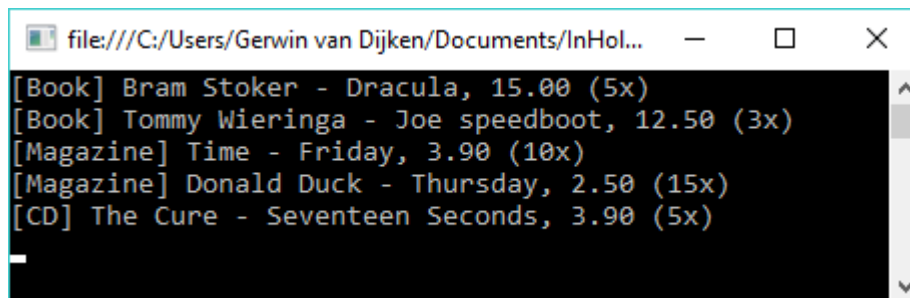
```
public string Title { get; set; }  
public float Price { get; set; }  
public int NumberOfItems { get; set; }
```

This information (title, price, count) is being passed to the constructor of class BookStoreItem.

Create 3 derived classes of BookStoreItem: Book, Magazine and CD. Give class Book an extra (`string`) property “Author”, class Magazine an extra (`DayOfWeek`) property “ReleaseDay”, and class CD an extra property (`string`) “Artist”.

Implement a class “BookStore” that contains a list of “BookStoreItem” objects. Make sure this list can only be accessed by the bookstore. Use a constructor to create the list. Add the next 2 methods to the BookStore class: “Add” (to add an item to the bookstore), “PrintAllItems” (to print all items in the bookstore).

Test the code by creating a bookstore object (in the main program), adding several books, magazines and CDs to it, and subsequently print all items. When printing the items, also the item-specific properties must be printed (like the author of a book, and the artist of a CD).



```
[Book] Bram Stoker - Dracula, 15.00 (5x)  
[Book] Tommy Wieringa - Joe speedboot, 12.50 (3x)  
[Magazine] Time - Friday, 3.90 (10x)  
[Magazine] Donald Duck - Thursday, 2.50 (15x)  
[CD] The Cure - Seventeen Seconds, 3.90 (5x)  
_
```

Try also to create an object of class “BookStoreItem”. What kind of error do you see?

Assignment 2

Create a new Console application and add file 'IFlyable.cs' to it, containing the following interface:

```
interface IFlyable
{
    void LiftOff();
    void Fly();
    void Land();
}
```

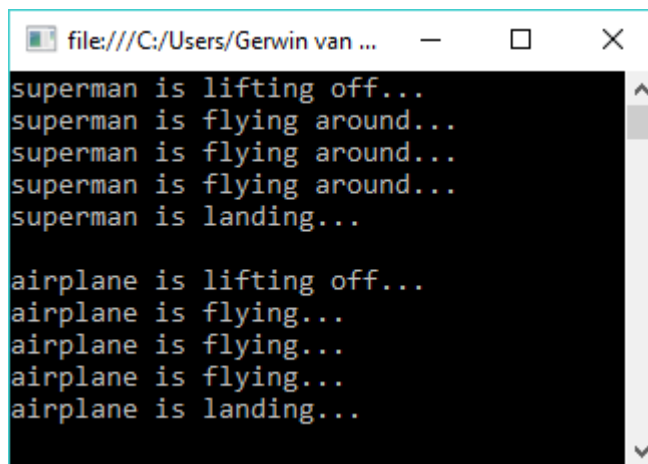
Now add 2 classes to the application: 'Superman' and 'Airplane', both implementing interface IFlyable.

Use the following code in the Main:

```
void Start(string[] args)
{
    IFlyable s = new Superman();
    IFlyable a = new Airplane();

    MakeFlight(s);
    MakeFlight(a);
}
```

and add a method 'MakeFlight' that receives a 'IFlyable' object, and calls the interface methods so the next output is generated (see screenshot below).



```
superman is lifting off...
superman is flying around...
superman is flying around...
superman is flying around...
superman is landing...

airplane is lifting off...
airplane is flying...
airplane is flying...
airplane is flying...
airplane is landing...
```

Assignment 3

Implement the next interfaces (each in a separate file):

```
public interface IPencil {
    bool CanWrite { get; } // determines if the pencil can still write
    void Write(string message); // writes characters of the message
    void AfterSharpening(); // the pencil is made 'new' (so it can write 'max' again)
}

public interface IPencilSharpener
{
    void Sharpen(IPencil pencil);
}
```

Create a class "Pencil" that implements interface "IPencil", and create a class "PencilSharpener" that implements interface "IPencilSharpener".

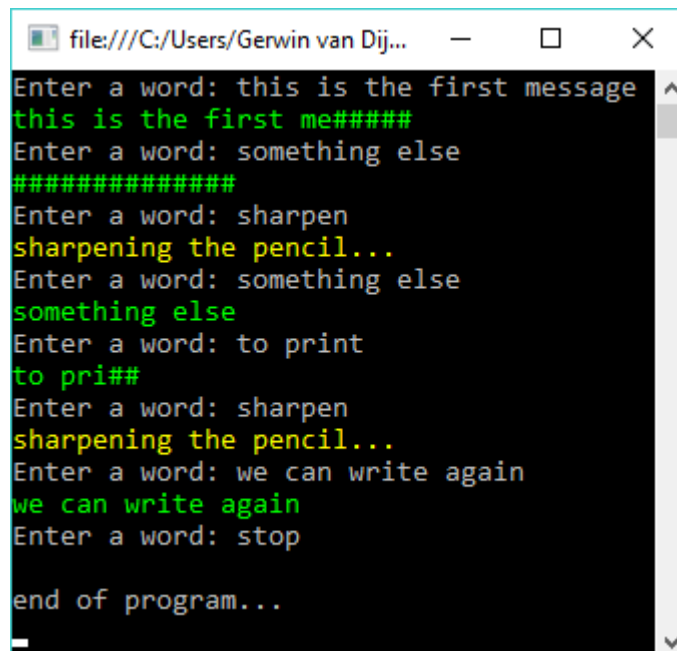
Add the following member fields to class Pencil:

```
private int maxToWrite; // number of characters to write with a sharpened pencil
private int nrOfCharsWritten; // number of written characters
```

The pencil will not be able to write the complete message if it's too long. After writing a certain number of characters, the pencil will become dull and each remaining character will be written as '#'.

Write a small program that reads messages until the user enters the word "stop". Each entered message should be written by the (created) pencil. If the user enters the word "sharpen", then the pencil must be sharpened by the (created) pencil-sharpener.

Output of the program could be like:



```
file:///C:/Users/Gerwin van Dij...
Enter a word: this is the first message
this is the first me#####
Enter a word: something else
#####
Enter a word: sharpen
sharpening the pencil...
Enter a word: something else
something else
Enter a word: to print
to pri##
Enter a word: sharpen
sharpening the pencil...
Enter a word: we can write again
we can write again
Enter a word: stop

end of program...
```